

Computer Architecture

What happens when you
Power On
a computer system?

All the way to being usable
by a human

When electric current shorts
flowing

Your system does "nothing"

The PROCESSOR enters
an infinite loop of doing
an operation NOP (on Intel systems)
No operation

What are our "modern" computers?

Digital

General Purpose computers

by opposition to a Specific purpose computer

Analogue vs Digital

Computers ?



Continuous
Data input
SIGNAL

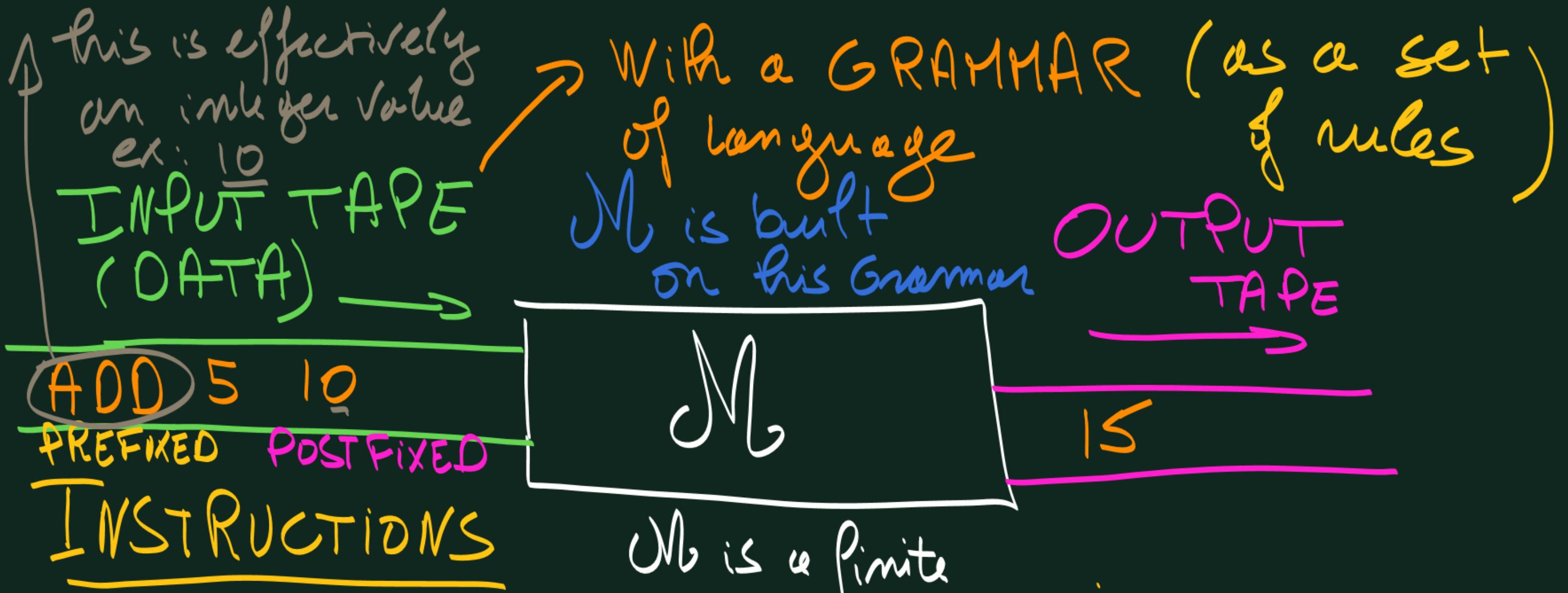


Discrete data
ex: Binary
Values

The Turing Machine

Pure mathematical construct
in Language Theory

What is a programmable Machine over integer
values?



M is a finite
 State Automaton

M is DETERMINISTIC
 For a given input, always some output

VALUES (apply
 Instructions)

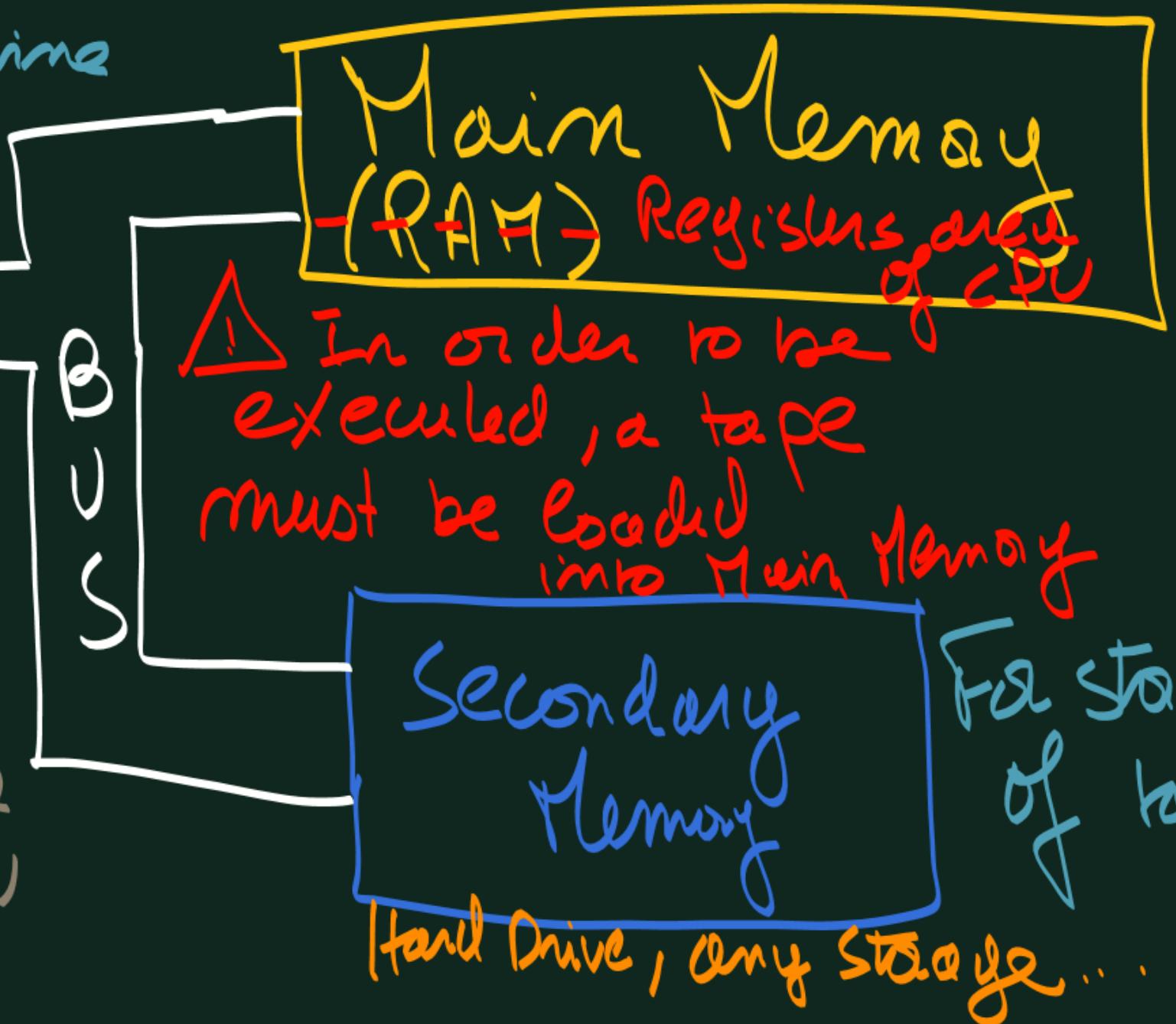
The Von Neumann Architecture

Finite State Machine
→ COMPUTER

Arithmetic & logic Unit

ALU
Part of CPU

One core
== One ALU



Turing Tape (PROGRAM) is in Main Memory
and Fed to ALU

for storage of tapes =

In order to execute a program, its entire code must be loaded in RAM

CPUs can only work with instructions and values in RAM

Turing Machine process the
tape SEQUENTIALLY

Instruction after instruction

So do our computers

At the rate of our computer's
per ALU clock. in GHz today

From the arrival of integrated Intel
microelectronics, in 70's Design

With Central Processing Unit (CPU)

chip which packages

ALU (at least one)

Bus connectivity (for data flow)

Memory Area (not RAM)

Random Access Memory (RAM) chips

We load programs (Code Data)
from which the CPU reads and writes

Fast Memory chips are expensive
to Manufacture

Input / Output bottleneck
Each component, including the BUS, has its own speed
and throughput

The Intel CPU Design implies

Only memory area where computations are performed
is INSIDE the CPU

FETCH
(from RAM)

DECODE
(apply grammar)

EXECUTE
(inside CPU)

WRITEBACK
(to RAM)

CPU Registers Area

Each CPU has its grammar

called the Instruction Set (IS)

Intel IS examples

Each instruction
is mapped
to integer
values
MICROCODE of CPU

{ ADD
MOV (move data)
NOP } Assembly
code

Intel < AMD
ARM
RISC-V (embedded
systems)

How do we get the CPU on
out of the NOP loop?
Or, how do we change the tape?

Task Manager

Type a name, publisher, or PID to search

Run new task

Details

Processes

Performance

App history

Startup apps

Users

Details

Services

System Idle Process

Iolaos.exe

System interrupts

System

ms-teams.exe

dwm.exe

TeamViewer/Desktop....

PPTService.exe

Taskmgr.exe

mongod.exe

MsMpEng.exe

FNPLicensingService6...

CPU doing NOP

Name	PID	Status	User name	CPU	Memory (act...)	Architect...	Description
System Idle Process	0	Running	SYSTEM	98	8 K	x86	Percentage of time the processor is idle
Iolaos.exe	11368	Running	RF	00	293,396 K	x86	Iolaos
System interrupts	-	Running	SYSTEM	00	0 K		Deferred procedure calls and interrupt service routines
System	4	Running	SYSTEM	00	20 K		NT Kernel & System
ms-teams.exe	2728	Running	RF	00	1,051,844 K	x64	Microsoft Teams (work or school)
dwm.exe	1428	Running	DWM-1	00	71,784 K	x64	Desktop Window Manager
TeamViewer/Desktop....	13472	Running	SYSTEM	00	779,708 K	x64	TeamViewer
PPTService.exe	14684	Running	RF	00	21,076 K	x86	PPTService
Taskmgr.exe	3404	Running	RF	00	55,512 K	x64	Task Manager
mongod.exe	5776	Running	NETWORK ...	00	276,752 K	x64	MongoDB Database Server
MsMpEng.exe	5524	Running	SYSTEM	00	292,352 K	x64	Antimalware Service Executable
FNPLicensingService6...	5264	Running	SYSTEM	00	2,612 K	x64	Activation Licensing Service

"Changing the tape" requires
someone or something to do it!

Welcome to the
Programmable
A chip
Hardware external, connected to CPU
and it can halt (interrupt) the CPU
and modify a particular CPU register

CPU registers

Two categories

The General/compile registers

Management registers

ex: where, in memory, is the
Next instruction I must execute?

Instruction Pointer Register (IP)

The PIC at
boot time

changes

the
IP register
value

A PIC has 16 to 32 areas of memory, each programmable to a value

The position of memory is hardwired (constant value in chip) to the memory location of the BIOS first instruction where ???

Read-Only Memory (ROM)

Non Writable (unlike RAM)

Non Volatile Memory (data remains
in memory
when no power)

Great tech for

Storing the first program a
computer must execute

⇒ Address Space 0 to N is unified between
ROM & RAM

So, the first program a computer must execute is at the first memory location of where the ROM starts where the Bios now UEFI ^{inst} instruction is (usually @ 0) address

Power On

↳ CPU goes to NOP

↳ Interruption Controller

Halts it an

Changes the CPU Instruction

Pointer register to

The address location of
first instruction of BIOS
(UEFI)

The Basic Input Output Subsystem
Bios

SOFTWARE

Handling of the System Bus

Bus: a system of electrical connections between components

No input /output otherwise

Once the Hardware(HW) integrity tests

Then, the Operating System must load

It's on Secondary Memory
(a storage device)

Must load in Main Memory *

↳ RAM ↳ CPU Registers

* ROM
RAM
CPU Registers

DECIDABILITY

Secondary Storage may have

0 to N
OSs

↳ We need a PROTOCOL
for this decision

The storage formatting process



The Master Boot Record (MBR) protocol

Sector 0 will be reserved to give
the sector address of the BOOTLOADER

Formatted into SECTORS
with each an address
Starting from 0
Each sector has a ~~fixed~~ size
Sector size is fixed at Formatting time

The code of the Boot Loader software
is now loaded in RAM

Windows
Apple OS
bootloaders do not have a
visible UI

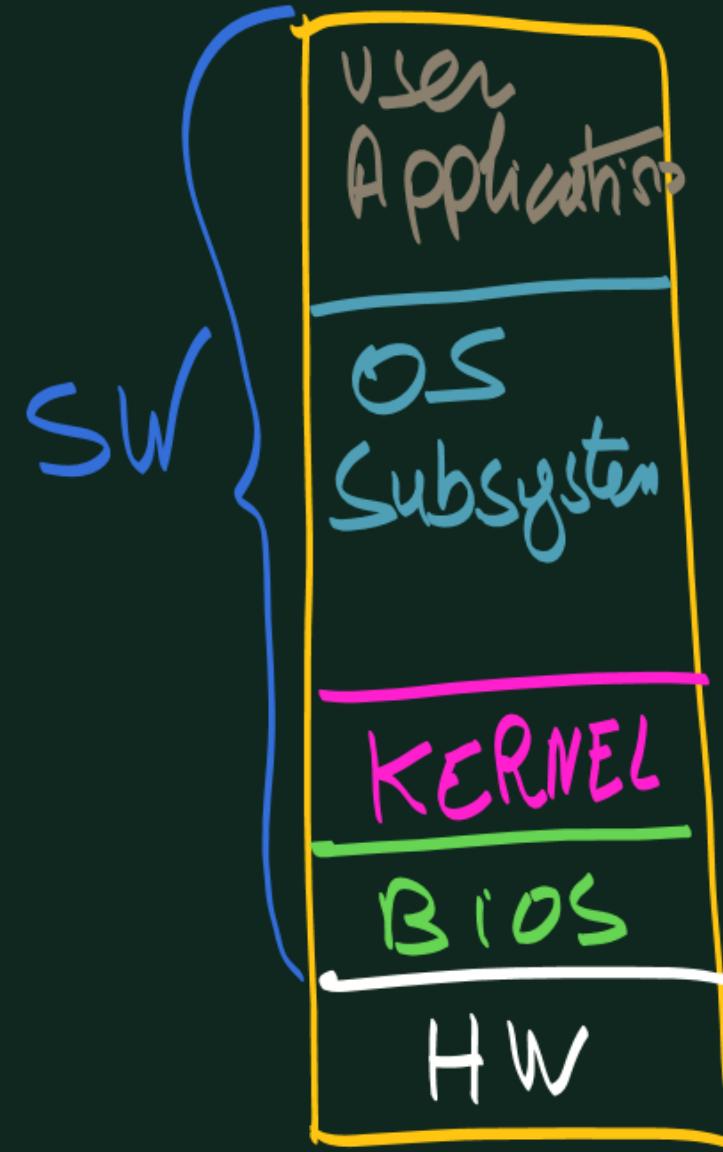
Linux one(s) does

Lilo
GRUB

UI with a user choice of OS to
be loaded

Where is the first sector
of the first component of target OS

The Operating System Kernel



The kernel of a "modern OS"
1970's : as per the UNIX Kernel

Kernighan & Ritchie

Creation of C Prog Lang to implement UNIX

- ① Hardware Appropriation
See and absolute owner of HW ressources
- ② Scheduling
- ③ Inter Process Communication (IPC)

What about I/O from/to devices?

Monolithic Kernel LINUX

Device drivers

code

run in Kernel
space

Performance ++

Device driver code

can jeopardise
the whole OS

Windows

HYBRID

Kernel

Some drivers

can run in

Kernel
mode

MacOS, iOS

Micro-Kernel

Device drivers

in User Space

With a lot of

IPC

Once loaded, the kernel loads
one component of the OS after another

The "Subsystem"

File Mngt

Networking

;

GUI

(Shell)

GUI

CLI

Once
the
shell is
loaded
the system
is usable

Back to CPU Registers

Their sizes must be known
The bitness of a CPU is fixed
32 bit registers \Rightarrow each register
Bitness has NOTHING to do with performance

Max value in 32 bit? $2^{32} \approx 4 \cdot 10^9$
(unsigned)

Max Value in 64 bit $2^{64} \approx 1.8 \cdot 10^{19}$
(unsigned)

A 32 bit system can address 4 GB of memory
64 bit system

64 bit system

WORD of Memory

8 bit CPU



Digital computers can only handle integers (YES!)

Only 0 or 1 in Computer Memory.

$$27_{10} \text{ in base } 2?$$

27 $\begin{array}{r} | \\ 2 \end{array}$
 | $\begin{array}{r} | \\ 13 \end{array}$
 | $\begin{array}{r} | \\ 2 \end{array}$
 | $\begin{array}{r} | \\ 6 \end{array}$
 | $\begin{array}{r} | \\ 2 \end{array}$
 | $\begin{array}{r} | \\ 0 \end{array}$
 | $\begin{array}{r} | \\ 3 \end{array}$
 | $\begin{array}{r} | \\ 2 \end{array}$
 | $\begin{array}{r} | \\ 1 \end{array}$
 | $\begin{array}{r} | \\ 0 \end{array}$ ← STOP

Real
Remainder
from Bottom
to top

$$27_{10} \rightarrow \text{base } 3$$

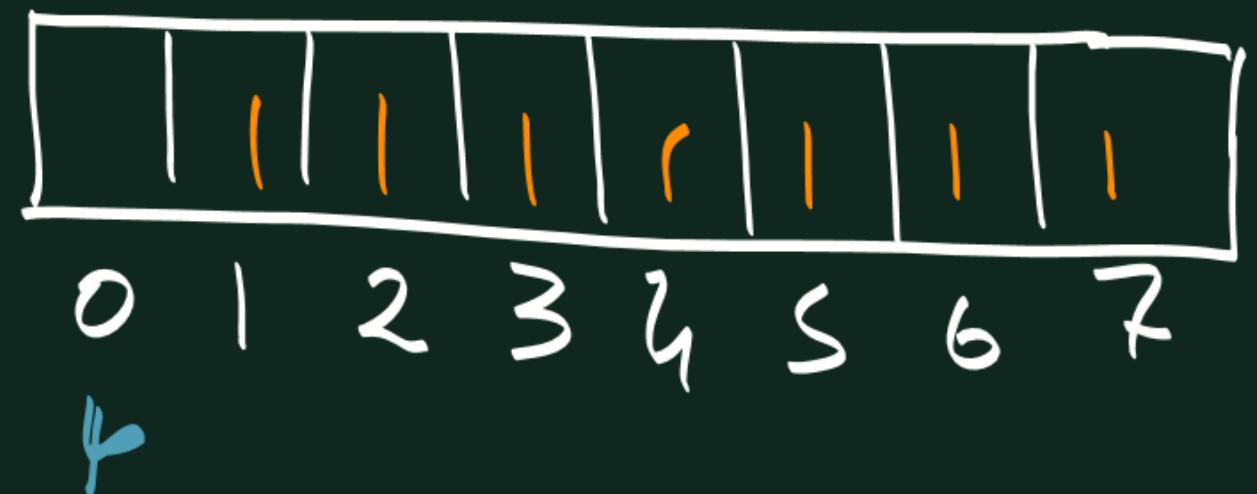
27 $\begin{array}{r} | \\ 3 \end{array}$
 | $\begin{array}{r} | \\ 9 \end{array}$
 | $\begin{array}{r} | \\ 0 \end{array}$
 | $\begin{array}{r} | \\ 3 \end{array}$
 | $\begin{array}{r} | \\ 0 \end{array}$
 | $\begin{array}{r} | \\ 3 \end{array}$
 | $\begin{array}{r} | \\ 0 \end{array}$ ← STOP

1000₃

1000₃ → base 10
 $1 \times 3^3 + 0 \times 3^2 + 0 \times 3^1 + 0 \times 3^0 = 27_{10}$

11011₂ → base 10
 $1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 16 + 8 + 0 + 2 + 1 = 27$

Suppose integer negative values
8 bit CPU



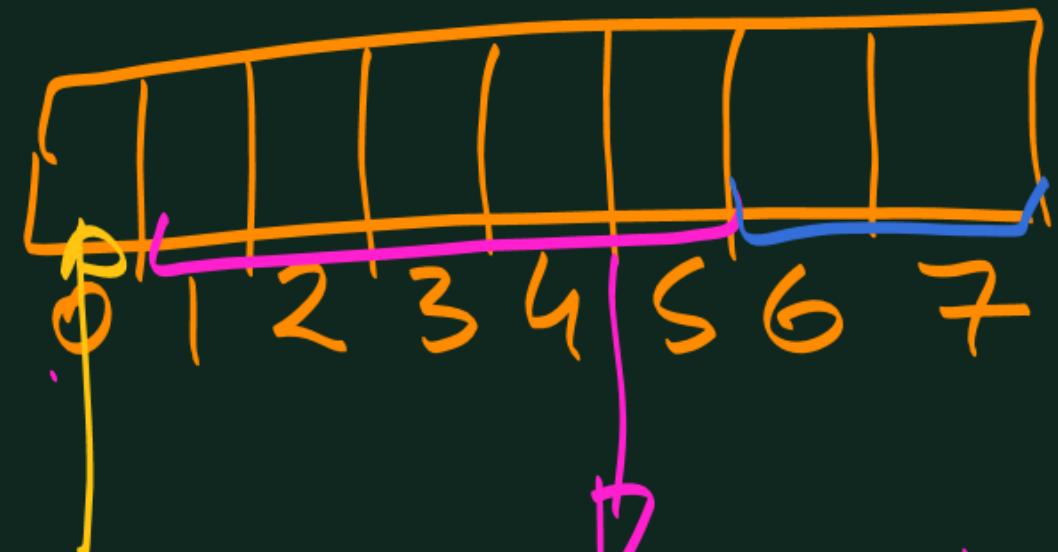
"Microscopic"
vision of CPU register

Sign bit
0 negative \Rightarrow Max value is 2^7 OK
1 positive WORD of Memory (in 8bit)
-128 ; 127



Real Numbers

CPU bitness
improves precision!



Sign bit

a Mantissa

ANY REAL NUMBER
ON A DIGITAL COMPUTER
IS AN APPROXIMATION

$a \cdot 10^b$ must be
integer

adjusted
to real
number

4.32 Exponent -2

(432) $\times 10^{-2}$

Mantissa

$$\begin{array}{c} \text{111}_2 \\ \rightarrow 2^3 + 2^2 + 2 + 1 \cdot 2^0 \\ = 8 + 4 + 2 + 1 = 15_{10} \end{array}$$

-15? \Rightarrow Assuming 4 bit CPU
 Two's complement

- ① Flip bits $\rightarrow 0000_2$
- ② Add 1 $\rightarrow 0001_2$

$$\begin{array}{r} 15 + (-15)? \\ \text{4 bit} \\ \text{CPU, east} \\ \text{Carry over is } + \\ \text{east(Overflow)} \end{array}$$

$$\begin{array}{r} 111 \\ + 0,0,0,1 \\ \hline 0000 \end{array}$$

$$\begin{array}{r} -15 + 1 \\ 0001 \\ + 0,0,0,1 \\ \hline 0010 \end{array}$$

0010 \Rightarrow -14
 2's complement

$$\begin{array}{r} 0FLIP \\ 0010 \\ + 1101 \\ \hline 11110 \end{array}$$

$$1 \times 2^3 + 1 \times 2^2 + 1 \times 2 + 0 \times 2^0 = 8 + 4 + 2 = 14$$

How are negative and positive values differentiated?

The "Most Significant bit" (MSB) is always

- 0 for positive values
- 1 for negative values

Convention
→ "One more on the negative side"

Example on 4 bit

Max unsigned value $2^4 = 16$
Range for unsigned [−8; 7] hence 16 values (inc 0)

7₁₀ → 0111₂
MSB

8₁₀ → 1000 (unsigned)
Two's complement → Flip 0111
+1
MSB | 000₂

Try and find the process
of launching a program
from the shell (Explorer
Finder)

How does the tape gets changed?
Notably with ONE ALU

Context

40 - 30 years ago

- ① One program runs at a time
- ② One ALU
- ③ One Simple Shell
You've just finished booking, the shell runs